ARMY RESEARCH LABORATORY

Detecting Planar Surfaces in Outdoor Urban Environments

by Philip David

ARL-TR-4599 **September 2008**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-4599 **September 2008**

Detecting Planar Surfaces in Outdoor Urban Environments

Philip David
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2008	Interim	October 2007 to August 2008

4. TITLE AND SUBTITLE
Detecting Planar Surfaces in Outdoor Urban Environments

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Philip David

5d. PROJECT NUMBER
8FEPMA

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRD-ARL-CI-IA
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-4599

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We describe an approach to automatically detect building facades in images of urban environments. This is an important problem in vision-based navigation, landmark recognition, and surveillance applications. In particular, with the proliferation of GPS- and camera-enabled cell phones, a backup geolocation system is needed when GPS satellite signals are blocked in so-called “urban canyons.”
Image line segments are first located, and then the vanishing points of these segments are determined using the RANSAC robust estimation algorithm. Next, the intersections of line segments associated with pairs of vanishing points are used to generate local support for planar facades at different orientations. The plane support points are then clustered using an algorithm that requires no knowledge of the number of clusters or of their spatial proximity. Finally, building facades are identified by fitting vanishing point-aligned quadrilaterals to the clustered support points. Our experiments show good performance in a number of complex urban environments. The main contribution of our approach is its improved performance over existing approaches while placing no constraints on the facades in terms of their number or orientation, and minimal constraints on the length of the detected line segments.

15. SUBJECT TERMS
Computer vision, vanishing point, building facade, planar surface

16. Security Classification of:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Philip David
U	U	U	UU	26	19b. TELEPHONE NUMBER (*Include area code*) (301) 394-5603

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

The ability to detect and recognize buildings is important to a variety of vision applications operating in outdoor urban environments. These include landmark recognition, assisted and autonomous navigation, image-based rendering, and 3D scene modeling. This report discusses a solution to one part of the building recognition problem, that of detecting multiple planar surfaces from a single image. Because each building facade can be described as a region of a scene plane at a specific position and orientation, the ability to generate a collection of building facades can be viewed as a first step in a system designed to solve any of the previously mentioned applications.

A number of general methods exist for scene surface recovery. The structure from motion approach is one of the most general *(1)*. From multiple images acquired from different viewpoints, the displacements of corresponding pixels from one image to the next are used to compute the 3D depth of the corresponding scene points. This depth information can then be segmented into qualitatively different surfaces by fitting parametric surfaces (e.g., planes and conics) to the depth values *(2)*. Werner and Zisserman use this approach for architectural model reconstruction from multiple images *(3)*. Liebowitz et al., discuss the same application, but use one or two images along with geometric constraints that are common to architectural scenes *(4)*.

In some cases, 3D properties of a scene must be inferred from a single image. For example, static surveillance cameras may be placed in urban environments at locations where Global Positioning System (GPS) signals cannot be received; in this case, accurate position and orientation of the camera relative to a world coordinate frame must be determined from a single perspective image of the environment.

Tourism is another application of single image structure recovery. The tourist of the near future will be able to point their camera-equipped mobile phone at the urban scene in front of them and ask questions such as *(5)*: Where am I? What building is this? How do I get to a particular location? These questions can be answered given the camera location and orientation, and given a 2D or 3D map of the environment. While GPS, which is now integrated into some mobile phones, could be used to determine location, in some urban environments tall building block GPS signals, rendering GPS unusable. Even when GPS can be received, it does not provide orientation, and position is only accurate to about 10 meters. Thus, vision-based location from a single image has the potential to increase the accuracy of information obtained from these mobile devices. Approaches to determine the orientation of a camera relative to the three dominant orthogonal directions in an urban scene include *(6,7,8)*.

Some approaches to recognizing planar surfaces from a single image assume the availability of 2D or 3D models that describe the facades of each building. A facade model may consist of an image of the facade, or of a collection of coplanar points or line segments. It is well known that images of a planar surface acquired from different viewpoints are related by a linear transformation known as a homography *(9)*. Given a model of a planar surface consisting of a set of point or line features, and a set of four or more corresponding features in an image from a calibrated camera, the position and orientation of the scene plane is uniquely determined from the homography relating the model and image [p. 213]*(10,9)* This geometric constraint may be used to find sets of image features lying on a common plane *(11,12)*.

Most existing single-view approaches that use building facades for navigation, recognition, etc., require that a single scene plane span the majority of the image. This enables straightforward matching between an image and a model. For example, Robertson and Cipolla *(12)* describe an approach to navigation in urban environments in which a single image acquired from a mobile phone is used to determine the position and orientation of the camera; they assume that the image is dominated by a single plane and match the query image to a database of facade images using correspondences of local color features centered on Harris corner points. When multiple planar surfaces are visible in an image, the image must be segmented into regions corresponding to each scene plane.

As any given image can be generated by an infinite number of 3D surfaces, when only a single image is available some assumptions about the geometric properties of the scene must be made in order to recover the surface geometry. Most urban building facades have surface markings due to doors, windows, bricks, and blocks. As such, each building facade generally consists of two sets of parallel lines, where lines in the first set intersect lines in the second set at right angles. It is well known that the perspective image of a collection of parallel scene edges intersect at a single point in the image, known as the vanishing point. Thus, the image of a building facade may be identified by locating regions in the image covered by pairs of intersecting edges, where each edge is oriented in the direction of one of two vanishing points. This is the approach that we take in this report.

Image line segments are first located, and then the vanishing points of these segments are determined using the RANSAC robust model fitting algorithm *(14)*. Groups of short segments are combined into longer segments while maintaining alignment with the associated vanishing points. Next, the intersections of line segments associated with pairs of vanishing points are used to generate local support for planar facades at different orientations. The plane support points are then clustered using an algorithm that requires no knowledge of the number of clusters or of their spatial proximity. Finally, building facades are identified by fitting vanishing point-aligned quadrilaterals to the clustered support points. The main contribution of our approach is its improved performance over existing approaches while

placing no constraints on the facades in terms of their number or orientation, and minimal constraints on the length of the detected line segments.

2. Related Work

Shape from texture and shading have been used in the past to estimate scene surface orientation. The shape from shading approach estimates the shape of a scene from a single image through the analysis of the gradual variation of shading in the image *(15)*. Shape from shading methods require the scene to consist of uniformly colored, Lambertian surfaces (these requirements allow the image brightness to be described as a function of surface shape and light source direction); this is not often the case in outdoor urban environments. Algorithms for shape from texture use the variation of texture primitives across an image to estimate the shape of the observed surface *(16)*. Most shape from texture algorithms are not useful for outdoor urban environments because they require the scene to consist of smooth surfaces with uniform texture *(17)*.

A variety of approaches to planar surface detection from a single image have been proposed in the past. Most of these approaches, however, make simplifying assumptions or require manual image segmentation by a user. A number of authors *(4,18,19,20)* have developed systems for 3D scene reconstruction from a single image where the user is required to manually identify image points and lines corresponding to coplanar or parallel scene points and lines. Sturm and Maybank *(18)* perform 3D reconstruction given user-provided coplanarity, perpendicularity, and parallelism constraints. Schaffalitzky and Zisserman *(19)* describe methods to detect image features that are the images of repeated patterns on world planes. These patterns include equally spaced coplanar parallel lines, elements repeated by translation in the plane, and elements arranged in a regular planar grid. The groupings are detected along with their vanishing points and lines, but the problem of automatically segmenting multiple planes in an image is not addressed. Liebowitz et al., *(4)* present methods to reconstruct piecewise planar objects from one or two views of a scene, but again, the problem of automatically segmenting multiple planes in an image is not addressed.

Hoiem et al., *(21)* propose an approach to computing coarse 3D geometric features of a scene from a single image. The coarse orientations of large surfaces in a scene are estimated by learning appearance-based models of surfaces at different orientations. The features used include color, texture, location, shape, and geometry of line segments. Image regions are classified as ground, sky, or vertical surface, with vertical surfaces subclassified into planar left, planar right, planar forward, nonplanar solid, and nonplanar porous.

The approach does a good job identifying vertical surfaces, but does not reliably identify the correct orientations of those surfaces.

Kosecka and Zhang *(22)* describe an approach to detecting building facades that relies on being able to detect a small number of long line segments along the borders of facades. This will be unreliable in many cluttered environments. Delage et al., *(23)* present an approach to 3D reconstruction of indoor environments from a single image using a calibrated camera whose height and orientation relative to the ground plane is assumed known; the floor-to-wall boundary is first identified using a Bayesian network, and then the 3D reconstruction is straightforward.

Our approach is similar to the approach of Micusik et al., *(24)* in which orthogonal planar surfaces are detected from a single image of an indoor environment. The orientation of each patch in an color-segmented image is determined by computing the maximum a posteriori (MAP) labeling in a Markov random field, where labels correspond to one of the three dominant orthogonal planes. Their approach was not applied to imagery of cluttered, outdoor environments, where building facades often consist of highly patterned, nonuniformly colored surfaces.

3. Detection of Vanishing Points

The majority of edges in an urban environment generally align with the three principle orthogonal directions of a local world coordinate frame. However, due to the presence of slanted surfaces (such as roofs), numerous edges at other orientations may also be present. But, the edges on any planar surface, whether slanted or not, are usually parallel or orthogonal to each other. Therefore, to detect all large planar surfaces, we locate the vanishing points of all large groups of parallel scene edges, regardless of their orientation.

Vanishing points have been used in the past to solve a number of calibration problems, including internal camera parameter estimation, relative orientation, image rectification, and object recognition. A variety of methods have been developed to detect and estimate vanishing points. Common approaches include image-space clustering *(25)*, the Hough transform *(26)*, and expectation maximization *(7)*. We use an approach based on the RANSAC robust model fitting algorithm *(14)* that is similar to the approach of Wildenauer and Vincze *(27)* .

The first step in our approach to detecting vanishing points is the detection of straight line segments. The Canny edge detector *(28)* with hysteresis thresholding is used to generate a binary image of edge points. Straight line segments are extracted from this edge image by first linking edges into contours and then splitting the contours into straight segments *(29)*.

The final line segments are those whose sum of squared distances to the contour points is minimized. Each line segment L_i is identified by its two endpoints: $L_i = \{(x_i^1, y_i^1), (x_i^2, y_i^2)\}$. In a 2048×1536 image, line segments shorter than 10 pixels are discarded. Figure 1 shows an image and the line segments detected in that image. This image will be used throughout sections 3 to 5 of this report to explain our approach.

Figure 1. Original image (left) and detected line segments (right).

For efficiency in computing the image vanishing points, for each line segment L_i, we precompute the normalized homogeneous representations of the coincident infinite line, $\mathbf{l_i}$, the endpoints, $\mathbf{e_i^1}$ and $\mathbf{e_i^2}$, and the midpoint, $\mathbf{m_i}$. These are calculated according to

$$e_i^1 = \left(x_i^1, y_i^1, 1\right)^T,$$

$$e_i^2, = \left(x_i^2, y_i^2, 1\right)^T,$$

$$m_i = \left(\left(x_i^1 + x_i^2\right)/2, \left(y_i^1 + y_i^2\right)/2, 1\right)^T,$$

$$l_i^{'} = e_i^1 \times e_i^2,$$

$$l_i = l_i^{'} / \sqrt{l_i^{'}(1)^2 + l_i^{'}(2)^2}.$$

The RANSAC algorithm is applied several times to the above data; each trial is used to locate the single vanishing point with the most support. Before each new trial, the data supporting the vanishing point found in the previous trail is removed. This process is repeated until V_{max} vanishing points are found, or until the size of the largest consensus set is less than S_{min}. (The values of these parameters and those that follow are given in section 6.) On each trial of RANSAC, T random samples of line pairs are examined. The line pair $\mathbf{l_i}$ and $\mathbf{l_j}$ seed a potential vanishing point $\mathbf{v_{ij}}$ when the segments L_i and L_j are each at least H_{seed} pixels long and when their angle, $\theta_{ij} = \cos^{-1}\left(\mathbf{l_i^1 l_j^1} + \mathbf{l_i^2 l_j^2}\right)$, is no larger than Θ_{seed}. The initial vanishing point of the line pair is $\mathbf{v_{ij}} = \mathbf{l_i} \times \mathbf{l_j}$. The normalized line through $\mathbf{v_{ij}}$

and the midpoint of line segment L_k is given by $\mathbf{l_{ijk}} = \mathbf{l'_{ijk}} / \sqrt{\mathbf{l'_{ijk}}(\mathbf{1})^2 + \mathbf{l'_{ijk}}(\mathbf{2})^2}$ where $\mathbf{l'_{ijk}} = \mathbf{v_{ij}} \times \mathbf{m_k}$. Then, line segment L_k is considered to support $\mathbf{v_{ij}}$ and is added to the consensus set C_{ij} when the perpendicular distance, $d_{ijk} = \mathbf{l_{ijk}} \times \mathbf{e_k^1}$, from one endpoint $\mathbf{e_k^1}$ of L_k to $\mathbf{l_{ijk}}$ is no larger than D_{sup} and when the angle between these lines, $\theta_{ijk} = \cos^{-1}\left(\mathbf{v_{ij}^1 l_{ijk}^1} + \mathbf{v_{ij}^2 l_{ijk}^2}\right)$, is no larger than Θ_{sup}. All line segments in the largest consensus set C_{max} are used to estimate the final location of the vanishing point, $\mathbf{v}^*$, for the current trial. $\mathbf{v}^*$ is required to minimize the weighted sum, for all lines $L_t \in C_{max}$, of the squared distances of line segment end points to the line through $\mathbf{v}^*$ and $\mathbf{m_t}$:

$$\mathbf{v}^* = \underset{\mathbf{v}}{\arg\min} \sum_{\mathbf{L_t} \in \mathbf{C_{max}}} \sqrt{(\mathbf{x_t^1} - \mathbf{x_t^2})^2 + (\mathbf{y_t^1} - \mathbf{y_t^2})^2} \left(\mathbf{l_{vt}} \times \mathbf{e_t^1}\right)^2$$

where $\mathbf{l_{vt}} = \mathbf{l'_{vt}} / \sqrt{\mathbf{l'_{vt}}(\mathbf{1})^2 + \mathbf{l'_{vt}}(\mathbf{2})^2}$ and $\mathbf{l'_{vt}} = \mathbf{v} \times \mathbf{m_t}$. $\mathbf{v}^*$ is found using standard methods for nonlinear optimization. After computing $\mathbf{v}^*$, all line segments $L_t \in C_{max}$ are corrected so that they are coincident with $\mathbf{v}^*$. The correction is performed by projecting the endpoints of each line segment L_t onto the line $\mathbf{l_{vt}^*} = \mathbf{v}^* \times \mathbf{m_t}$ through $\mathbf{v}^*$ and $\mathbf{m_t}$. Figure 2 shows the line segments from the example image classified by the vanishing point that each supports.

Figure 2. Line segments from the example image classified by vanishing point.

4. Detection of Consistent Clusters of Plane Support

Image line segments that have been labeled according to vanishing point provide an initial cue to segmenting planar regions in the image. Under the assumption that intersecting edges in the scene are coplanar and orthogonal, every pair of nearby, nonparallel, vanishing point-aligned image line segments defines the local surface orientation of the scene point that projects to the segment intersection point in the image. For two local image regions to be images of the same plane, the pairs of intersecting line segments in each of the two regions should be labeled with the same two vanishing points. We therefore seek to cluster pairs of intersecting line segments that have identical vanishing point label pairs.

Not all pairs of vanishing points define the orientation of a plane that can be easily detected in an image. Vanishing directions that are close to parallel correspond to planes that are highly forshortened: their normals are nearly perpendicular to the camera line of sight, and their image consists of line pairs that are nearly parallel and very dense. These line segments will be very difficult to accurately detect. Although building facades may occur at these orientations, what is more common is that two nearly parallel vanishing directions correspond to edges on two different, nonparallel planes. Hence, to label the intersections of lines aligned with a pair of vanishing points, we require that the mean angle between their pairs of intersecting line segments be sufficiently large. In all experiments reported here, these angles were required to be in the range $45° - 135°$.

For each pair of vanishing points, $(\mathbf{v_i}, \mathbf{v_j})$, we find all points of intersection between pairs of line segments where one segment is aligned with $\mathbf{v_i}$ and the other segmentis aligned with $\mathbf{v_j}$. Only line segments that are spatially close in the image, and with no other segments in between, are allowed to generate intersection points. One cannot simply examine the segments whose endpoints are close, as an intersection point of two segments may be near the center of one of the segments. Like most line segment detection algorithms, ours produces non-intersecting segments. To detect intersections of segments that approach but do not meet (at a corners or at a T-junctions), we first extend the ends of all segments by D_{ext} pixels. Then, a straightforward approach to locating intersection points is to consider all pairs of line segments. However, for high-resolution images such as ours (2048×1536), there are often 5000 or more line segments in an image. Checking on the order of 5000^2 pairs of line segments for intersections is a computationally expensive procedure. Instead, we create a line segment index image by rasterizing the line segments. The index k of each extended line segment L_k is recorded at each pixel in the index image over which segment L_k passes; multiple indices may be recorded at any pixel. Then, the index image is searched for pixels at which two or more indices have been recorded. If indices k and m are recorded at the same pixel in the index image, and one of L_k or L_m aligns with $\mathbf{v_i}$ and the other with $\mathbf{v_j}$, then $\mathbf{p} = \mathbf{l_k} \times \mathbf{l_m}$ (the exact intersection of segments L_k and L_m) is recorded

as a plane support point with label (i, j). This process allows all line segment intersections to be found in time linear in the number of segments. Figure 3 shows the set of plane support points for the image shown in figure 1.

The labeled plane support points define local regions in the image that support planes of various orientations. We seek maximal clusters of similarly labeled support points. These clusters define the largest spatial regions in the image that may correspond to a single plane (a building facade) in the scene.

Note that multiple scene planes with the same orientation, corresponding to parallel but distinct building facades, will be assigned the same labels. Separating these identically labeled support points into regions corresponding to separate scene planes is one goal of the clustering process described next. The other goal of clustering is to remove spurious support points. In general, the support points for one plane should not lie inside a cluster of support points for a different plane. However, in most real images, intersecting line segments occur that do not correspond to orthogonal edges in the scene. These are due to spurious and non-orthogonal line segments detected on planar surfaces, as well as line segments detected on non-planar objects such as trees, vehicles, clouds, etc. The spurious plane support points generated by these segments can occur anywhere in an image, including in the interior region of a cluster of support points for a true plane.

If two parallel scene planes are to be detected as separate planes, the support points for the two planes must group into spatially separate clusters in the image. However, as shown in figure 4, spatial separation is not a sufficient condition. The two clusters of identically

Figure 3. Plane support points for all pairs of vanishing points with sufficiently large mean angle between segments.

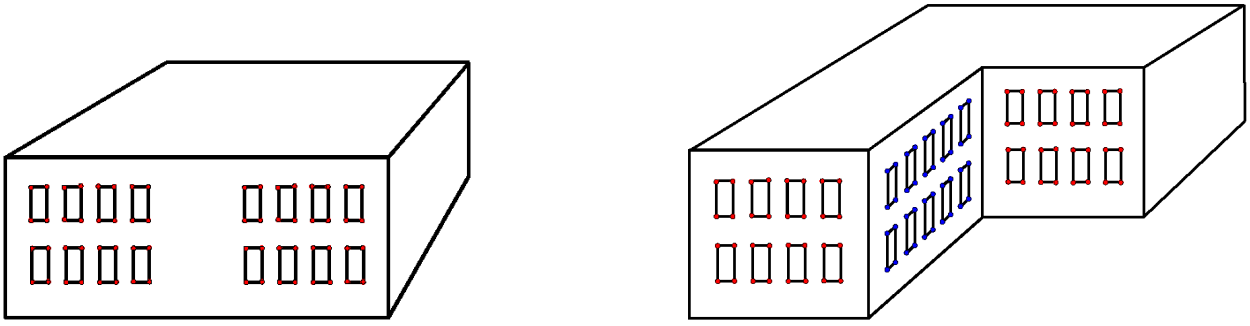

Figure 4. (a) Spatial separation of two clusters of plane support points with the same label is insufficient to infer two separate building facades. (b) The two clusters must also be divided by support points of a plane at some other orientation.

labeled support points must also be divided by the support points of a plane at some other orientation. To carry out this clustering, a nonsymmetric $N \times N$ binary adjacency matrix A is created where N is the total number of plane support points (for all labels). We set $A_{i,j} = 1$ to indicate that support point $\mathbf{p_i}$ is allowed to be grouped with the cluster that includes support point $\mathbf{p_j}$; otherwise, $A_{i,j} = 0$. Given A, a symmetric adjacency matrix A' of the same size is created: if points $\mathbf{p_i}$ and $\mathbf{p_j}$ each agree to be joined to the others cluster, i.e., $A_{i,j} = 1$ and $A_{j,i} = 1$, then $A'_{i,j} = A'_{j,i} = 1$. Finally, the connected components of A' are found from the Dulmage-Mendelsohn matrix decomposition *(30)* of A'. These connected components are the clusters of plane support points that define the building facades.

It remains to define when a support point $\mathbf{p_i}$ is allowed to be grouped with the cluster that includes support point $\mathbf{p_j}$. The values in row i of matrix A are assigned in order of increasing distance from $\mathbf{p_i}$: first column j_1, then column j_2, ..., and finally column j_N, where $\|\mathbf{p_i} - \mathbf{p_{j_1}}\| \leq \|\mathbf{p_i} - \mathbf{p_{j_2}}\| \leq \cdots \leq \|\mathbf{p_i} - \mathbf{p_{j_N}}\|$. Note that for all i, $j_1 = i$ and $A_{i,i} = 1$. For the remaining columns, A_{ij_k} is assigned a value of 1 only if the orientation of the vector $\overrightarrow{\mathbf{p_i p_{j_k}}}$ is in the range of angles from $\mathbf{p_i}$ that does not include any previous support points $(\mathbf{p_{j_1}}, \mathbf{p_{j_2}}, \ldots, \mathbf{p_{j_{k-1}}})$ whose labels are different from that of $\mathbf{p_i}$. More specifically, let $label\,(\mathbf{p_j})$ denote the label assigned to support point $\mathbf{p_j}$ and let $\vartheta\,(\mathbf{v})$ denote the orientation of vector $\mathbf{v}$. Define

$$\theta^k_{min}\,(\mathbf{p}_i) = \min_{\theta}\ \theta \geq \vartheta\left(\overrightarrow{\mathbf{p}_i\mathbf{p}_{j_t}}\right) \text{ for all } \mathbf{p}_{j_t} \text{ where } label\,(\mathbf{p}_{j_t}) \neq label\,(\mathbf{p}_i)\,,\ 1 \leq t \leq k-1, \quad (1)$$

$$\theta^k_{max}\,(\mathbf{p}_i) = \max_{\theta}\ \theta \geq \vartheta\left(\overrightarrow{\mathbf{p}_i\mathbf{p}_{j_t}}\right) \text{ for all } \mathbf{p}_{j_t} \text{ where } label\,(\mathbf{p}_{j_t}) \neq label\,(\mathbf{p}_i)\,,\ 1 \leq t \leq k-1. \quad (2)$$

Then,

$$A_{i,j_k} = 1 \text{ iff } \vartheta\left(\overrightarrow{\mathbf{p}_i\mathbf{p}_{j_k}}\right) \in \left[\theta^k_{min}\,(\mathbf{p}_i)\,,\ \theta^k_{max}\,(\mathbf{p}_i)\right]. \quad (3)$$

Figure 5 illustrates this process.

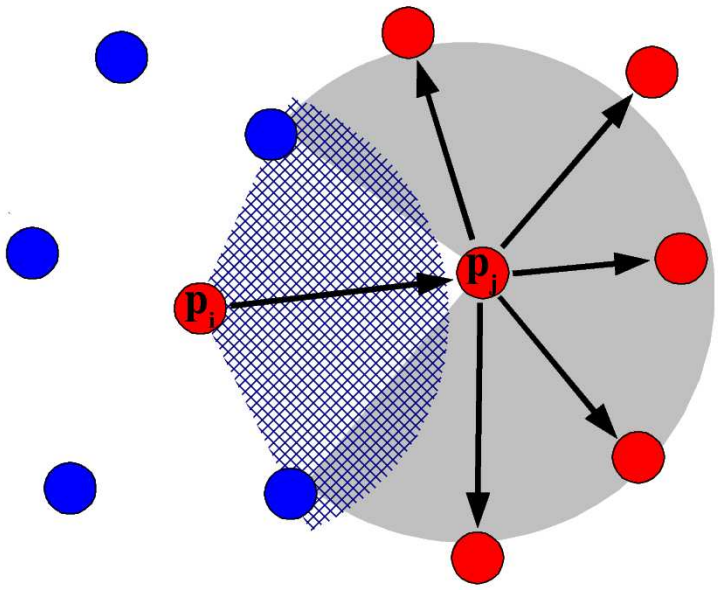

Figure 5. Calculation of support point adjacency. Support point $\mathbf{p_j}$ is adjacent to $\mathbf{p_i}$ (i.e., $A_{i,j} = 1$) because $\mathbf{p_j}$ is inside the largest arc centered at $\mathbf{p_i}$ (indicated by the cross hatching) which includes only red support points. However, support point $\mathbf{p_i}$ is not adjacent to $\mathbf{p_j}$ (i.e., $A_{j,i} = 0$) because $\mathbf{p_i}$ is outside the largest arc centered at $\mathbf{p_j}$ (indicated by the gray shading) which includes only red support points.

A number of optimizations to speed up this process are possible. One is to check the adjacency of a support point only if the direction to that point differs from all previous points by more than some threshold (i.e., $5 - 10°$). Also, a limit on the number of points checked or on the distance to points may be used to end the process early. Figure 6 shows the connected components of the adjacency matrix A' generated for the example image.

5. Fitting Quadrilaterals to Plane Support Clusters

As building facades are almost always rectangular, and because the image of a rectangle is a quadrilateral, we next fit quadrilaterals to the clusters of plane support points. The clusters of plane support points, defined by the connected components, usually provide a good estimate of the regions in an image corresponding to different scene planes. However, occasional clustering errors do occur. The clustering errors that have the largest impact on the accuracy of detected facades are those that occur near the cluster boundaries. Many of these clustering errors can be corrected by smoothing the boundaries of the clusters. This is most easily accomplished by first rasterizing each connected component graph, that is, by creating an image of the arcs connecting the nodes in the graph, and then applying the

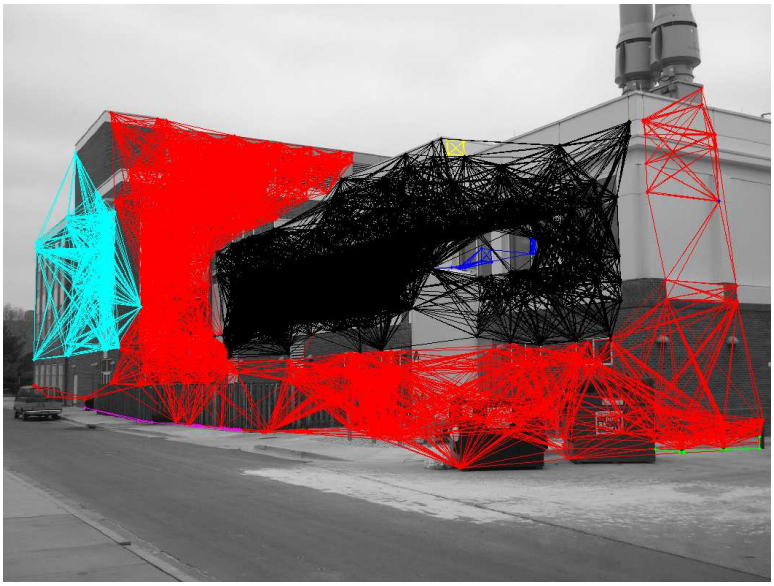

Figure 6. Initial connected components of plane support points for the example image.

mathematical morphology operations of erosion and dilation to this image. To reduce the occurrence of holes in the rasterized graph in dense regions of the graph, the image of the graphs are created at a resolution that is a multiple of R_{dec} of the original image's resolution. Then, the morphological operations can be applied to this image. First the rasterized graph is eroded using a circular structuring element of radius R_{erode} pixels, then the blob with the largest area is dilated with a circular structuring element of size $R_{erode} + 1$. Given the smoothed rasterized image of a cluster, the final cluster is the set of support points in the original cluster which lie inside the smoothed image of that cluster. Figure 7 illustrates the process of smoothing a cluster of support points and figure 8 shows all of the smoothed clusters for the example image.

The final step in locating building facades is to fit a quadrilateral to the convex hull of each cluster of plane support points. We assume that all building facades are rectangular, and

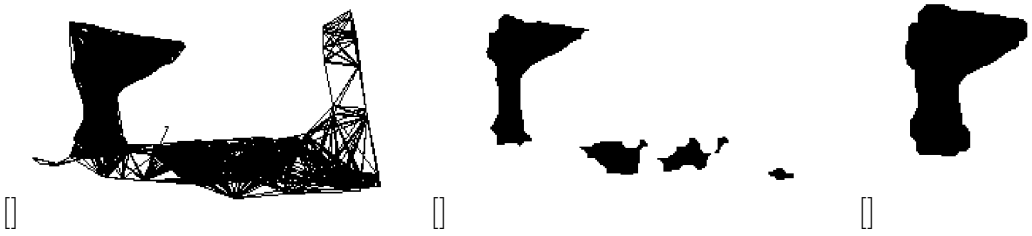

Figure 7. Smoothing a cluster of plane support points. (a) Rasterized adjacency graph for one cluster of support points. (b) Eroded image of adjacency graph. (c) Smoothed region of plane support.

Figure 8. The final smoothed clusters of plane support points for the example image.

assume that the boundaries of each facade are parallel to one of the two dominant orientations of edges on the surface of the facade. Therefore, opposite edges of a facade quadrilateral are required to align with one of the two vanishing points associated with the point cluster. We determine the smallest quadrilateral that encloses the point cluster's convex hull such that each edge of the quadrilateral passes through one vanishing point and one point on the convex hull of the cluster. When a vanishing point is finite, the two tangent lines making up the opposite edges of the bounding quadrilateral are easily found by scanning through all points on the cluster's convex hull, and locating those lines through the vanishing point and the hull point that make the smallest and largest angles with respect to the line from the vanishing point to the cluster centroid. When a vanishing point is at infinity, the distance of hull points from the line through the centroid is used to determine the tangent lines. Figure 9 shows the quadrilaterals corresponding to the building facades.

Figure 9. Building facades are determined by the vanishing point-aligned quadrilaterals that bound each smoothed cluster of support points.

6. Experiments

Figure 10 shows additional examples of using our algorithm to detect building facades in urban environments. As shown in these and the previous experiments, we obtain good results on images of a number of complex buildings. As seen in figure 10, not all of the final clusters of plane support points correspond to true building facades. Some clusters correspond to building roofs, some to reflections of building facades in windows, and some clusters correspond to walls inside of buildings. These false facades can easily be filtered out based on their small size when compared to the larger facades that are detected.

The values of the parameters used in our experiments are $V_{max} = 5$, $S_{min} = 20$, $T = 50$, $H_{seed} = 15$, $\Theta_{seed} = 40°$, $D_{sup} = 3$ pixels, $\Theta_{sup} = 3°$, $D_{ext} = 4$ pixels, $R_{dec} = 0.125$, and $R_{erode} = 4$. Although there are a significant number of parameters, we have found it easy to set them so as to obtain good performance. Furthermore, the performance of our algorithm is not highly sensitive to their values as small changes do not significantly affect the results.

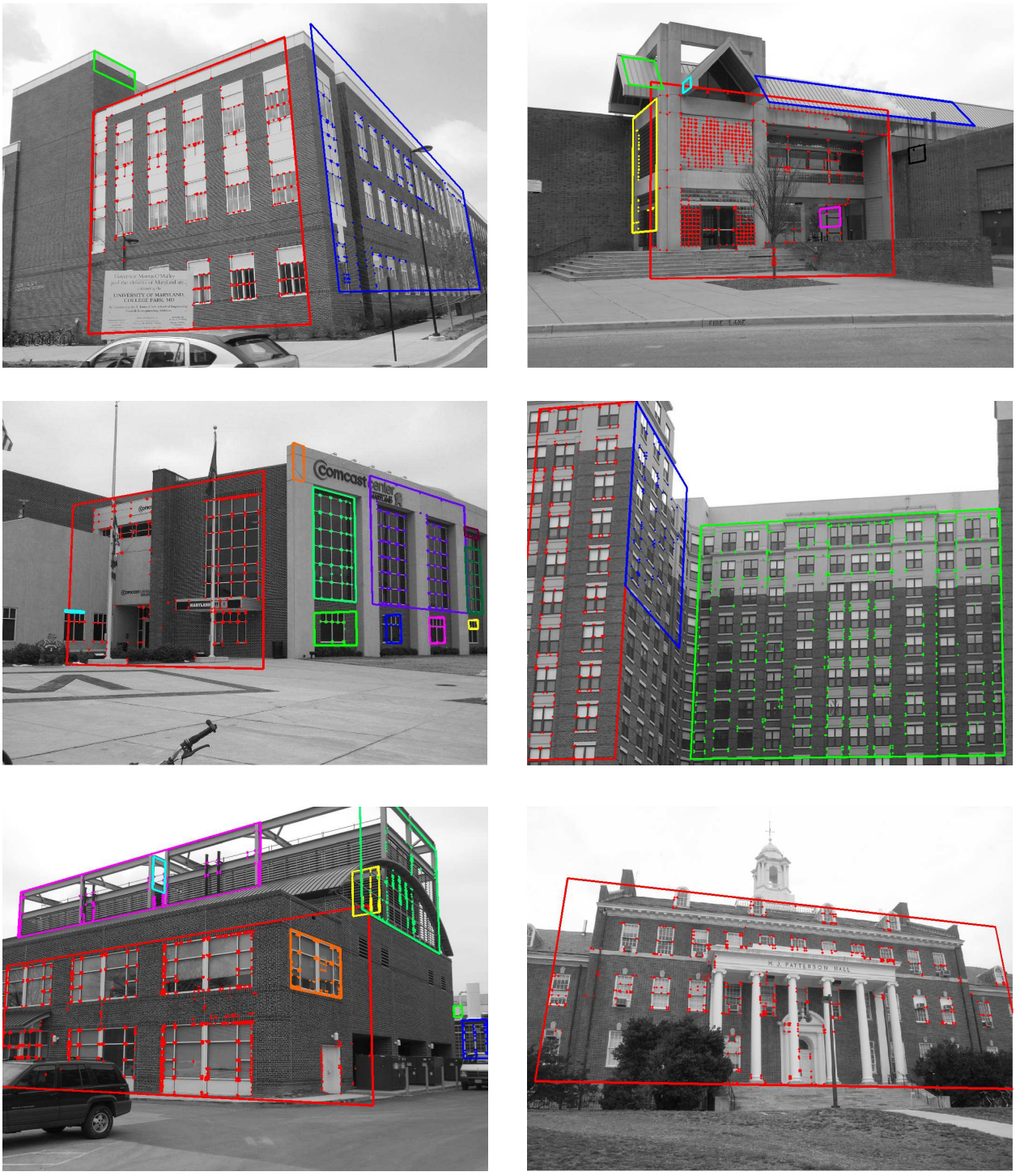

Figure 10. Building facades (and their support points) detected in other urban scenes.

7. Conclusions

We have demonstrated how a small amount of knowledge about the structure of an urban environment can be used to effectively locate multiple planar building facades from a single image. The main advantages of our approach over existing approaches are its improved performance in complex environments, the lack of a requirement for a single facade to be dominant in the image, and the ability to detect facades even when clutter makes it difficult to detect the line segments that form the facade boundaries. Our initial experiments show that the algorithm has good performance on a number of difficult scenes. In the future, we will investigate alternate clustering algorithms, which may require fewer parameters, and will investigate the use of other sources of information such as color and texture. Additional experiments will be conducted to test the algorithm's performance in a larger variety of urban environments. We also plan to integrate this facade detection algorithm into a system for building recognition and autonomous navigation in urban environments.

References

[1] Shindler, G.; Krishnaurthy, P.; Dellaert, P. Line-Based structure from motion for urban environments. in *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, **June 2006**, 846–853.

[2] Feddema, J.; Little, C. Rapid world modelig fitting range data to geometric primitives. in *Proc. IEEE Int. Conf. on Robotics and Automation*, **April 1997**, *4*, 2807–2812.

[3] Werner, T.; Zisserman, A. New techniques for automated architectural reconstructionfrom photographs. in *Proc. European Conference on computer Vision*, **May 2002**, pp. 541–555.

[4] Liebowitz, D.; Criminisi, A.; Zasserman, A. Creating architectural models from images. *EuroGraphics* **1999**, *18* (3), 39–50.

[5] http://geovector.com, GeoVector Corporation, Retrieved January 2008.

[6] Coughlan, J.; Yuille, A. Manhattan world: Compass direction froma single image by bayesian inference. in *Proc. IEEE Int. Conf on Computer Vision*, **1999**, 941–947.

[7] Antone, M.; Teller, S. Automatic recovery of relative camera rotations for urban scenes. in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, **June 2000**, *II*, 282–289.

[8] Kosecka, J.; Zhang W. Video compass. in *Proc. European conference on Computer Vision*, **May 2002**, 657–673.

[9] Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press 2 ed, April 2004.

[10] Strum, P. Algorithms for plane-based pose estimation. in *Proc. IEEE Conf. on Computer Vision and Patter Recognition* **June 2000**, *1*, 706–711.

[11] Lourakis, M.; Argyros, A.; Orphanoudakis, S. Detecting planes in an uncalibrated image pair. in *Proc. British Machine Vision Conference* **September 2000**, *2*, 587–596.

[12] Shindler, K. Generalized use of homographics for piecewise planar reconstruction. in *Proc 13th Scandinavian Conference on Image Analysis*, **June 2003**, 470–476.

[13] Robertson, D.; Cipolla, R. An image-based system for urban navigation. in *British Machine Vision Conference*, **September 2004**.

[14] Fischler, M. A.; Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm . Association for computing Machinery* **June 1981**, *24*, 381–395.

[15] Zhang Zhang, R.; Tsai, P.-S.; Cryet, J.; Shah, M. Shape-from-shading: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **August 1999**, *21*, 690–706.

[16] Kanatani, K.; Chou, T. C. Shape from texture: general principle. *Artifician Intelligence* **February 1989**, *38*, 1–48.

[17] Francos, J.; Permuter, H. Parametric estimation of the orientation of textured planar surfaces. *IEEE Trans. on Image Processing* **March 2001**, *10*, 403–418.

[18] Sturm, P. F.; Maybank, S. J. A method for intactive 3d reconstruction of piercewise planar objects from single images. in *BMVC*, **1999**, 265–274

[19] Schaffalitzky, F.; Zisserman, A. Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing* **June 2000**, *18*, 647–658.

[20] Li, Z.; Liu, J.; Tang, X. A closed-form solution to 3d reconstruction of piecewise planar objects from single images. in *CVPR*, **June 2007**.

[21] Hoiem, D.; Efros, A.; Hebert, M. Geometric context from a single image. in *Proc. IEEE Int. Conf. on Computer Vision*, **October 2005**, *1*, pp. 654–661.

[22] Kosecka, J.; Zhang, W. Extraction, matching and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding* **December 2005**, *100*, 274–293.

[23] Delage, E.; Lee, H.; Ng, A. Y. A dynamic Bayesian network model for autonomous 3d reconstruction from a signle indoor image. in *CVPR*, **2006**, *2*, 2418–2428.

[24] Miscusik, H.W.B.; Vincze, M. Towards detection of orthogonal planes in monocular images of indoor enviornments. in *Proc. IEEE Int. Conf. on Robotics and Automation*, **May 2008**, 999.

[25] McLean, G.; Kotturi, D. Vanishing point detection by line clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **November 1995**, *17*, 1090–1095.

[26] Magee, M.; Aggarwal, J. Determining vanishing points from perspective images. *computer Vision Graphics and Image Processing* **May 1984**, *26*, 256–267.

[27] Wildenauer, H.; Vincze, M. Vanishing point detection in complex man-made worlds. in *Proc In. Conf. on Image Analysis and Processing*, **September 2007**, 615–622.

[28] Canny, J. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **November 1986**, *8*.

[29] Kovesi, P. D. MATLAB and Octgave functions for computer vision and image processing. School of Computer Science & Software Engineering. The University of Wester Australia. Available from:
<http://www.csse.uwa.edu.au/ ∼ pk/research/matlabfns/>.

[30] Dulmage, A.; Mendelsohn, N. Coverings of bipartite graphs. *Canadian Journal of mathematics* **1958**, *10*, 517–534.

No. of Copies	Organization
1 (ELECT COPY)	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	DARPA ATTN IXO S WELBY 3701 N FAIRFAX DR ARLINGTON VA 22203-1714
1	OFC OF THE SECY OF DEFNS ATTN ODDRE (R&AT) THE PENTAGON WASHINGTON DC 20301-3080
1	US ARMY RSRCH DEV AND ENGRG CMND ARMAMENT RSRCH DEV AND ENGRG CTR ARMAMENT ENGRG AND TECHNLGY CTR ATTN AMSRD AAR AEF T J MATTS BLDG 305 ABERDEEN PROVING GROUND MD 21005-5001
1	US ARMY TRADOC BATTLE LAB INTEGRATION & TECHL DIRCTRT ATTN ATCD B 10 WHISTLER LANE FT MONROE VA 23651-5850
1	US ARMY INFO SYS ENGRG CMND ATTN AMSEL IE TD F JENIA FT HUACHUCA AZ 85613-5300
1	COMMANDER US ARMY RDECOM ATTN AMSRD AMR W C MCCORKLE 5400 FOWLER RD REDSTONE ARSENAL AL 35898-5000

No. of Copies	Organization
1	US GOVERNMENT PRINT OFF DEPOSITORY RECEIVING SECTION ATTN MAIL STOP IDAD J TATE 732 NORTH CAPITOL ST NW WASHINGTON DC 20402
1	US ARMY RSRCH LAB ATTN AMSRD ARL CI OK TP TECHL LIB T LANDFRIED BLDG 4600 ABERDEEN PROVING GROUND MD 21005-5066
1	DIRECTOR US ARMY RSRCH LAB ATTN AMSRD ARL RO EV W D BACH PO BOX 12211 RESEARCH TRIANGLE PARK NC 27709
11	US ARMY RSRCH LAB ATTN AMSRD ARL CI I B BROOME ATTN AMSRD ARL CI IA P DAVID (6 COPIES) ATTN AMSRD ARL CI IA S H YOUNG ATTN AMSRD ARL CI OK PE TECHL PUB ATTN AMSRD ARL CI OK TL TECHL LIB ATTN IMNE ALC IMS MAIL & RECORDS MGMT ADELPHI MD 20783-197
TOTAL::	21 (20 HCs 1 ELECT)

INTENTIONALLY LEFT BLANK